1. Classic American names

Photo by Travis Wise on Wikimedia (https://commons.wikimedia.org/wiki/File:Hello_My_Name_Is_(15283079263).jpg).

How have American baby name tastes changed since 1920? Which names have remained popular for over 100 years, and how do those names compare to more recent top baby names? These are considerations for many new parents, but the skills we'll practice while answering these queries are broadly applicable. After all, understanding trends and popularity is important for many businesses, too!

We'll be working with data provided by the United States Social Security Administration, which lists first names along with the number and sex of babies they were given to in each year. For processing speed purposes, we've limited the dataset to first names which were given to over 5,000 American babies in a given year. Our data spans 101 years, from 1920 through 2020.

`baby_names`

column	type	meaning
`year`	int	year
`first_name`	varchar	first name
`sex`	varchar	`sex` of babies given `first_name`
`num`	int	number of babies of `sex` given `first_name` in that `year`

Let's get oriented to American baby name tastes by looking at the names that have stood the test of time!

In [96]:
```
%%sql
postgresql:///names
select first_name, sum(num) as sum
from baby_names
group by first_name
having count(distinct year) = 101
order by sum desc
```

8 rows affected.

Out[96]:

first_name	sum
James	4748138
John	4510721
William	3614424
David	3571498
Joseph	2361382
Thomas	2166802
Charles	2112352
Elizabeth	1436286

2. Timeless or trendy?

Wow, it looks like there are a lot of timeless traditionally male names! Elizabeth is holding her own for the female names, too.

Now, let's broaden our understanding of the dataset by looking at all names. We'll attempt to capture the type of popularity that each name in the dataset enjoyed. Was the name classic and popular across many years or trendy, only popular for a few years? Let's find out.

In [98]:
```
%%sql
select first_name, sum(num) as sum, case when count(first_name) > 80 then 'C
            when count(first_name) > 50 then 'Semi-classic'
            when count(first_name) > 20 then 'Semi-trendy'
            else 'Trendy' end as popularity_type
from baby_names
group by first_name
order by first_name
```

```
 * postgresql:///names
547 rows affected.
```

Out[98]:

first_name	sum	popularity_type
Aaliyah	15870	Trendy
Aaron	530592	Semi-classic
Abigail	338485	Semi-trendy
Adam	497293	Semi-trendy
Addison	107433	Trendy
Adrian	147741	Semi-trendy
Aidan	68566	Trendy
Aiden	216194	Trendy
Alan	162041	Semi-trendy
Albert	260945	Semi-trendy

3. Top-ranked female names since 1920

Did you find your favorite American celebrity's name on the popularity chart? Was it classic or trendy? How do you think the name Henry did? What about Jaxon?

Since we didn't get many traditionally female names in our classic American names search in the first task, let's limit our search to names which were given to female babies.

We can use this opportunity to practice window functions by assigning a rank to female names based on the number of babies that have ever been given that name. What are the top-ranked female names since 1920?

In [100]:
```
%%sql
select rank() over(order by sum(num) desc) as name_rank, first_name, sum(num
from baby_names
where sex = 'F'
group by first_name
limit 10
```

```
 * postgresql:///names
10 rows affected.
```

Out[100]:

name_rank	first_name	sum
1	Mary	3215850
2	Patricia	1479802
3	Elizabeth	1436286
4	Jennifer	1404743
5	Linda	1361021
6	Barbara	1343901
7	Susan	1025728
8	Jessica	994210
9	Lisa	920119
10	Betty	893396

4. Picking a baby name

Perhaps a friend has heard of our work analyzing baby names and would like help choosing a name for her baby, a girl. She doesn't like any of the top-ranked names we found in the previous task.

She's set on a traditionally female name ending in the letter 'a' since she's heard that vowels in baby names are trendy. She's also looking for a name that has been popular in the years since 2015.

Let's see what we can do to find some options for this friend!

In [102]:
```
%%sql
select first_name
from baby_names
where sex = 'F' and year > 2015 and first_name like '%a'
group by first_name
order by sum(num) desc
```

```
 * postgresql:///names
19 rows affected.
```

Out[102]:

first_name
Olivia
Emma
Ava
Sophia
Isabella
Mia
Amelia
Ella
Sofia
Camila
Aria
Victoria
Layla
Nora
Mila
Luna
Stella
Gianna
Aurora

5. The Olivia expansion

Based on the results in the previous task, we can see that Olivia is the most popular female name ending in 'A' since 2015. When did the name Olivia become so popular?

Let's explore the rise of the name Olivia with the help of a window function.

In [104]:
```
%%sql
select year, first_name, num, sum(num) over(order by year) as cumulative_oli
from baby_names
where first_name = 'Olivia'
```

```
 * postgresql:///names
30 rows affected.
```

Out[104]:

year	first_name	num	cumulative_olivias
1991	Olivia	5601	5601
1992	Olivia	5809	11410
1993	Olivia	6340	17750
1994	Olivia	6434	24184
1995	Olivia	7624	31808
1996	Olivia	8124	39932
1997	Olivia	9477	49409
1998	Olivia	10610	60019
1999	Olivia	11255	71274
2000	Olivia	12852	84126
2001	Olivia	13977	98103
2002	Olivia	14630	112733
2003	Olivia	16152	128885
2004	Olivia	16106	144991
2005	Olivia	15694	160685
2006	Olivia	15501	176186
2007	Olivia	16584	192770
2008	Olivia	17084	209854
2009	Olivia	17438	227292
2010	Olivia	17029	244321
2011	Olivia	17327	261648
2012	Olivia	17320	278968
2013	Olivia	18439	297407
2014	Olivia	19823	317230
2015	Olivia	19710	336940
2016	Olivia	19380	356320
2017	Olivia	18744	375064
2018	Olivia	18011	393075
2019	Olivia	18508	411583
2020	Olivia	17535	429118

6. Many males with the same name

Wow, Olivia has had a meteoric rise! Let's take a look at traditionally male names now. We saw in the first task that there are nine traditionally male names given to at least 5,000 babies every single year in our 101-year dataset! Those names are classics, but showing up in the dataset every year doesn't necessarily mean that the timeless names were the most popular. Let's explore popular male names a little further.

In the next two tasks, we will build up to listing every year along with the most popular male name in that year. This presents a common problem: how do we find the greatest X in a group? Or, in the context of this problem, how do we find the male name given to the highest number of babies in a year?

In SQL, one approach is to use a subquery. We can first write a query that selects the `year` and the maximum `num` of babies given any single male name in that year. For example, in 1989, the male name given to the highest number of babies was given to 65,339 babies. We'll write this query in this task. In the next task, we can use the code from this task as a subquery to look up the `first_name` that was given to 65,339 babies in 1989… as well as the top male first name for all other years!

In [106]:
```
%%sql
select year, max(num) as max_num
from baby_names
where sex = 'M'
group by year
order by year
```

```
 * postgresql:///names
101 rows affected.
```

Out[106]:

year	max_num
1920	56914
1921	58215
1922	57280
1923	57469
1924	60801
1925	60897
1926	61130
1927	61671
1928	60703
1929	59804

7. Top male names over the years

In the previous task, we found the maximum number of babies given any one male name in each year. Incredibly, the most popular name each year varied from being given to less than 20,000 babies to being given to more than 90,000!

In this task, we find out what that top male name is for each year in our dataset.

In [108]:
```
%%sql
select b.year, b.first_name, b.num
from baby_names as b
inner join (select year, max(num) as max_num
            from baby_names
            where sex = 'M'
            group by year) as subquery
on b.year = subquery.year and b.num = subquery.max_num
order by year desc
```

```
 * postgresql:///names
101 rows affected.
```

Out[108]:

year	first_name	num
2020	Liam	19659
2019	Liam	20555
2018	Liam	19924
2017	Liam	18824
2016	Noah	19154
2015	Noah	19650
2014	Noah	19319
2013	Noah	18266
2012	Jacob	19088
2011	Jacob	20378

8. The most years at number one

Noah and Liam have ruled the roost in the last few years, but if we scroll down in the results, it looks like Michael and Jacob have also spent a good number of years as the top name! Which name has been number one for the largest number of years? Let's use a common table expression to find out.

In [110]:
```
%%sql
with cte as (select b.year, b.first_name, b.num
from baby_names as b
inner join (select year, max(num) as max_num
            from baby_names
            where sex = 'M'
            group by year) as subquery
on b.year = subquery.year and b.num = subquery.max_num
order by year desc)

select first_name, count(first_name) as count_top_name
from cte
group by first_name
order by count(first_name) desc
```

```
 * postgresql:///names
8 rows affected.
```

Out[110]:

first_name	count_top_name
Michael	44
Robert	17
Jacob	14
James	13
Noah	4
John	4
Liam	4
David	1